

# COMP 3111 Project Team T-33 Activity 1

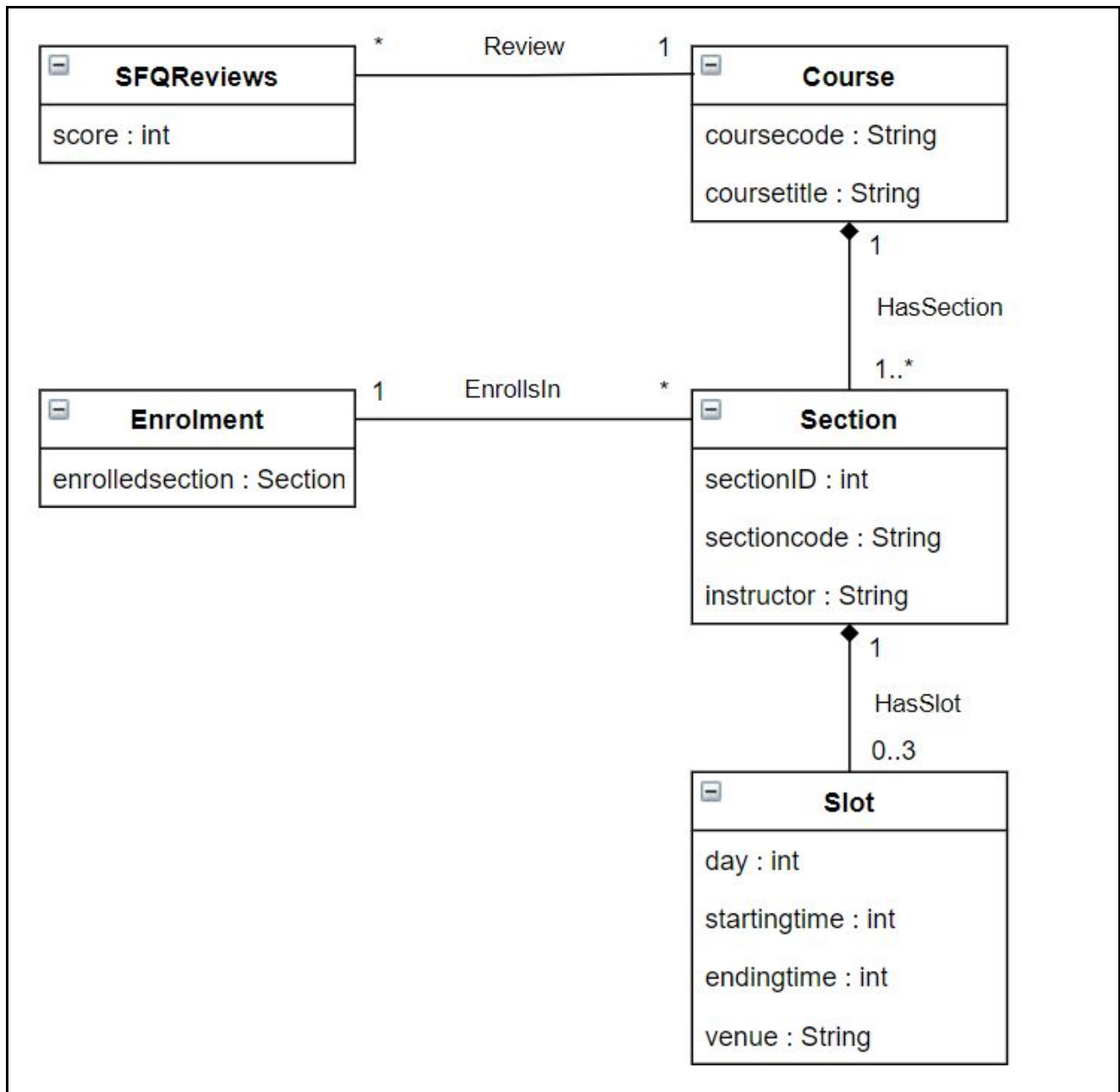
## Members

Kong Ming Hin	(mhkongaa)	Task 2 & Task 3
Pang Pak Long	(plpang)	Task 1 & Task 4
Ho Kin Chun	(kchoao)	Task 5 & Task 6

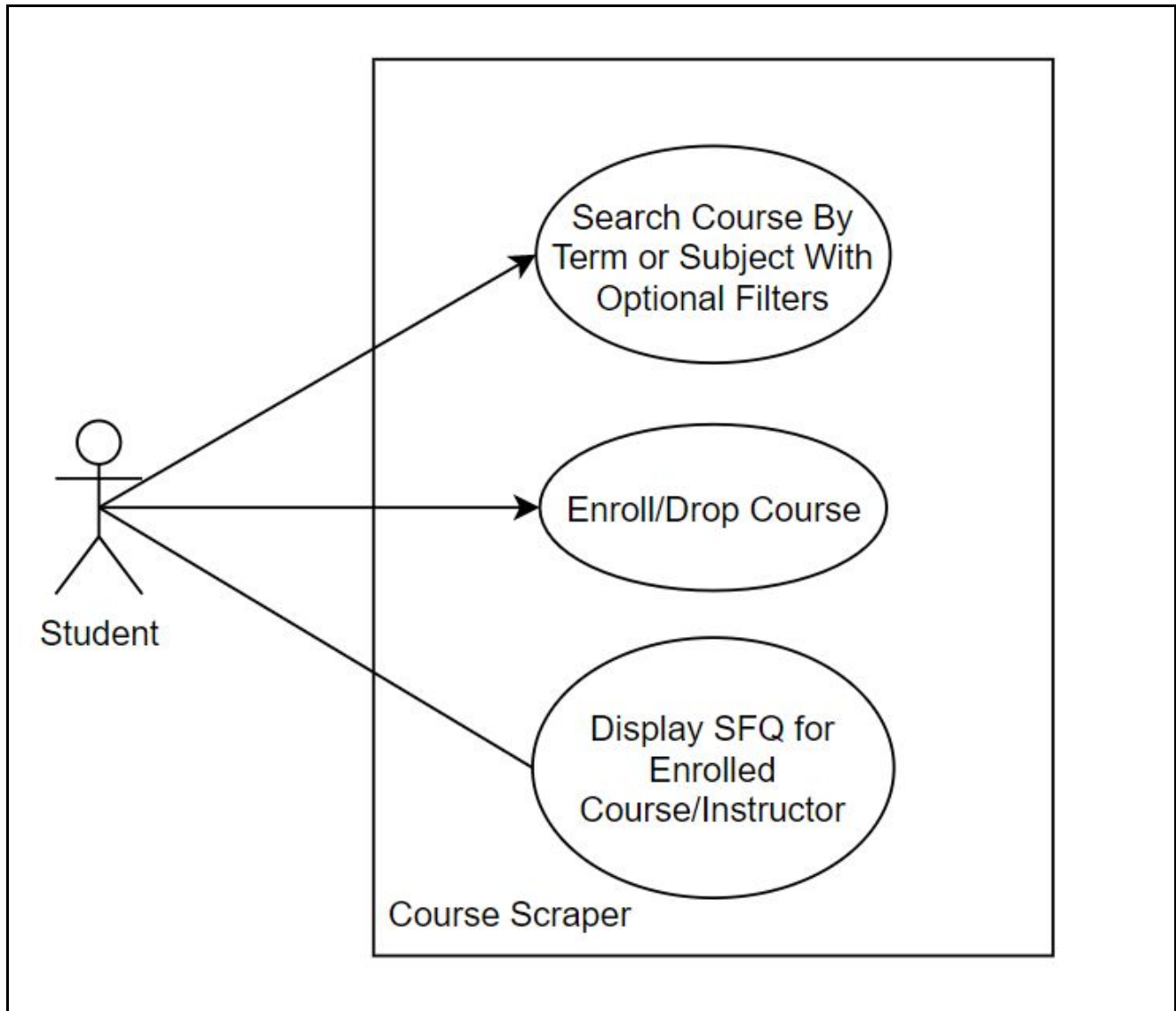
## Contents

Data Model Diagram	2
Use Case Diagram	3
Use Case 1 Detailed Specification (Task 1, 2, 3, 5)	4
Use Case 2 Detailed Specification (Task 3 & 4)	6
Use Case 3 Detailed Specification (Task 6)	7

## Data Model Diagram



## Use Case Diagram

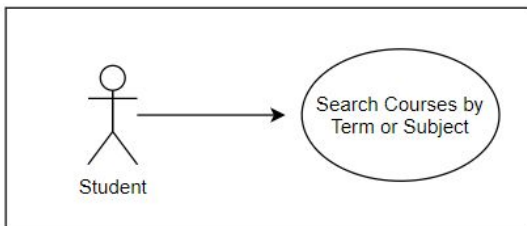


## Use Case 1 Detailed Specification (Task 1, 2, 3, 5)

### Brief Description

This use case describes how a student can use the system to search for courses. Students can either search by term and subject or search by term only. Students can apply specific filters when searching if needed.

### Diagram



### Preconditions

1. Students must specify term and subject information before using the basic search.

### Basic Flow

1. The use case begins when the student selects the Main tab or the Filter tab.
  - 1.1. If the Main tab is selected.
    - 1.1.1. The system displays the interface for the student to enter term and subject information.  
{Enter Term}  
{Enter Subject} (Optional)
      - 1.1.1.1. If Subject is empty, the basic search function will be disabled.
    - 1.1.2. The student indicates the term and subject in which he/she would like to search.  
{Search}
    - 1.1.3. The system fetches the course according to the term and subject, along with any optional filters. Display the result in the List tab.
    - 1.1.4. The system displays additional information, including the number of different sections, total number of courses and instructor who has a teaching assignment this term but does not need to teach at Tu 3:10 pm.
  - 1.2. If the Filter tab is selected.
    - 1.2.1. The system displays the interface for the student to select search filters.  
{Select Filters}
      - 1.2.1.1. Filters are applied. The student can go to the Main tab to continue the flow, or continue to interact with the filters.
2. The use case ends

## Alternative Flows

### A1: Invalid Term or Subject

At {Enter Term} or {Enter Subject} if the entered term or subject is invalid,

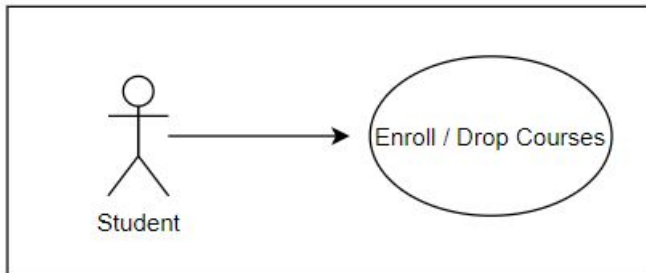
1. The system informs the student whether the term or subject is invalid
2. The flow of events is resumed at {Enter Term} or {Enter Subject}

## Use Case 2 Detailed Specification (Task 3 & 4)

### Brief Description

This use case describes how a student can enroll or drop a course using the system.

### Diagram



### Preconditions

1. Use Case 1 is executed successfully: the system already fetched and displayed the search result.

### Basic Flow

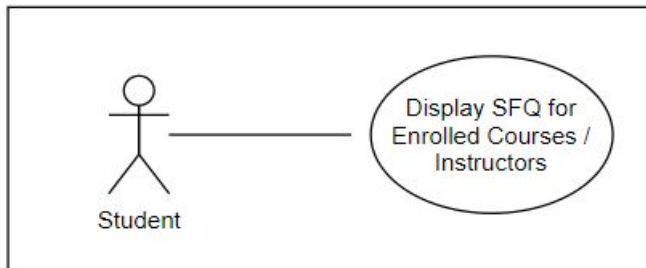
1. The use case begins when the Student actor decide to modify or review his / her enrolment
  - 1.1. If the Student actor selects the List tab.
    - 1.1.1. The student selects a course to enroll/drop.
      - 1.1.1.1. If the course selected is already enrolled.
        - 1.1.1.1.1. Drop the selected course.
      - 1.1.1.2. If the course selected is not enrolled.
        - 1.1.1.2.1. Enroll the selected course.
    - 1.1.2. The system updates the timetable whenever there is enrollment status change.
    - 1.1.3. The student can continue selecting another course to enroll by repeating step 1.1.1.
  - 1.2. If the Student actor selects the Time Table tab.
    - 1.2.1. The system fetches and displays all the enrolled sections as a distinct colour block in the table.
    - 1.2.2. If time clashes happen.
      - 1.2.2.1. The system overlaps two sections in the time clashed slot.
      - 1.2.2.2. The overlapped area will be displayed in a different colour.
2. The use case ends

## Use Case 3 Detailed Specification (Task 6)

### Brief Description

This use case describes how a student can find the SFQ with his/her enrolled course or the instructors' SFQ.

### Diagram



### Preconditions

1. Use Case 1 is executed successfully: the system already fetched and displayed the search result.

### Basic Flow

1. The use case begins when the Student actor decides to find the SFQ with his/her enrolled course or the instructors' SFQ.
  - 1.1. If the Student actor selects Find SFQ with my enrolled courses.
    - 1.1.1. If the course has no multiple sections.
      - 1.1.1.1. The system prints unadjusted SFQ data.
    - 1.1.2. If a course has multiple sections
      - 1.1.2.1. The system sums up the score from all sections and divides by the number of sections.
      - 1.1.2.2. The system prints average unadjusted SFQ data.
      - 1.1.2.3. If the List is empty, enroll the first 5 sections of the scraped data and jump to the step 1.1.2.
      - 1.1.2.4. If less than 5 sections, enroll all sections and jump to the step to 1.1.2.
  - 1.2. If the Student actor selects List instructors' average SFQ.
    - 1.2.1. The system searches for all instructors' data and their unadjusted SFQ score.
      - 1.2.1.1. If an instructor has taught only one section/course.
        - 1.2.1.1.1. The system prints instructors' names and his/her unadjusted SFQ score.
      - 1.2.1.2. If an instructor has taught more than one section/course.
        - 1.2.1.2.1. The system sums up all unadjusted SFQ scores of the sections taught by him/her and divides them by the number of sections.
        - 1.2.1.2.2. The system prints all instructors' names and his/her average SFQ score.
2. The use case ends.